

1 CLAIMS

2
3 1. One or more processor-accessible media comprising processor-
4 executable instructions that, when executed, direct a device to perform actions
5 comprising:

6 determining if an instruction of a line of common intermediate language
7 code meets a predetermined exception-related criterion; and

8 if so, injecting a decision point in association with the instruction of the line
9 of common intermediate language code, the decision point enabling a decision as
10 to whether an exception is to be thrown with respect to the instruction.

11
12 2. The one or more processor-accessible media as recited in claim 1,
13 comprising the processor-executable instructions that, when executed, direct the
14 device to perform further actions comprising:

15 retrieving the line of common intermediate language code from a common
16 intermediate language code program prior to the determining;

17 retrieving another line of common intermediate language code from the
18 common intermediate language code program; and

19 repeating the action of determining and the action of injecting for an
20 instruction of the other retrieved line of common intermediate language code.

1 3. The one or more processor-accessible media as recited in claim 1,
2 wherein the action of determining comprises an action of:

3 determining if the instruction of the line of common intermediate
4 language code is capable of throwing an exception.

5
6 4. The one or more processor-accessible media as recited in claim 1,
7 wherein the action of determining comprises an action of:

8 determining if the instruction of the line of common intermediate
9 language code is capable of throwing an exception and is related to a pre-
10 selected exception area.

11
12 5. The one or more processor-accessible media as recited in claim 1,
13 wherein the action of determining comprises an action of:

14 determining if the instruction of the line of common intermediate
15 language code is capable of throwing an exception with reference to a
16 common intermediate language code specification.

17
18 6. The one or more processor-accessible media as recited in claim 1,
19 wherein the action of injecting comprises an action of:

20 injecting a bookmark entry.

1 7. The one or more processor-accessible media as recited in claim 6,
2 wherein the action of injecting a bookmark entry comprises an action of:

3 injecting an instruction type indicator that indicates an
4 instruction type for the instruction of the line of common
5 intermediate language code.

6
7 8. The one or more processor-accessible media as recited in claim 6,
8 wherein the action of injecting a bookmark entry comprises an action of:

9 injecting an identifier that uniquely identifies the decision
10 point within the common intermediate language code that is being
11 instrumented.

12
13 9. The one or more processor-accessible media as recited in claim 1,
14 wherein the action of injecting comprises an action of:

15 injecting a call to a decision runtime library, the decision runtime
16 library comprising a program that is adapted to evaluate whether the
17 exception is to be thrown with respect to the instruction.

1 10. The one or more processor-accessible media as recited in claim 1,
2 wherein at least a portion of the processor-executable instructions comprise at
3 least part of an instrumentation tool that produces instrumented common
4 intermediate language code from the common intermediate language code by
5 repeating the actions of determining and injecting for a plurality of respective
6 instructions of a plurality of respective lines of the common intermediate language
7 code.

8
9 11. The one or more processor-accessible media as recited in claim 1,
10 wherein the one or more processor-accessible media comprise at least one of (i)
11 one or more storage media and (ii) one or more transmission media.

12
13 12. A device comprising:
14 instrumented common intermediate language code that includes a test
15 couplet corresponding to a decision point and an associated instruction, the
16 associated instruction capable of causing a fault;

17 a decision runtime library that is adapted to evaluate the test couplet to
18 selectively decide whether to throw an exception with respect to the associated
19 instruction; and

20 a common language runtime component that interprets the decision point so
21 as to call the decision runtime library prior to executing the associated instruction.

22
23 13. The device as recited in claim 12, wherein the instrumented
24 common intermediate language code is in a binary form.

1 14. The device as recited in claim 12, wherein the decision point
2 comprises a bookmark entry and a call to the decision runtime library, and wherein
3 the bookmark entry comprises an indication of an instruction type of the
4 associated instruction and an identifier of the decision point.

5
6 15. The device as recited in claim 14, wherein the decision runtime
7 library is further adapted to evaluate the test couplet to selectively decide whether
8 to throw an exception responsive to the bookmark entry.

9
10 16. The device as recited in claim 14, wherein the decision runtime
11 library is further adapted to evaluate the test couplet to selectively decide whether
12 to throw an exception responsive to the bookmark entry and based on throw
13 exception decision logic.

14
15 17. The device as recited in claim 14, wherein the decision runtime
16 library is further adapted to evaluate the test couplet to selectively decide whether
17 to throw an exception responsive to the bookmark entry and based on at least one
18 throw exception decision logic factor selected from the group comprising:
19 throwing an exception randomly, throwing an exception when first encountering a
20 given decision point using an identifier of the given decision point, and throwing
21 an exception when encountering a particular decision point along each new
22 execution path using an identifier of the particular decision point and one or more
23 stack values.

1 **18.** The device as recited in claim 14, wherein the decision runtime
2 library is further adapted to evaluate the test couplet to selectively decide whether
3 to throw an exception responsive to the indication of the instruction type of the
4 associated instruction.

5
6 **19.** The device as recited in claim 12, wherein the decision runtime
7 library is (i) modularized by exception category and/or (ii) operative in
8 dependence on an instruction type of the associated instruction.

9
10 **20.** An arrangement for enabling reliability testing of managed code, the
11 arrangement comprising:

12 instrumentation means for instrumenting common intermediate language
13 code with a plurality of decision points to produce instrumented common
14 intermediate language code; and

15 decision means for deciding whether to throw an exception at each decision
16 point of the plurality of decision points.

1 **21.** The arrangement as recited in claim 20, wherein the instrumentation
2 means comprises:

3 analysis means for analyzing whether individual instructions of a
4 plurality of instructions of the common intermediate language code can
5 result in a failure; and

6 injection means for injecting a respective decision point in
7 association with each respective individual instruction, which can result in
8 a failure as analyzed by the analysis means, of the plurality of instructions
9 of the common intermediate language code.

10
11 **22.** The arrangement as recited in claim 21, wherein the injection means
12 comprises:

13 means for injecting a respective bookmark entry that indicates
14 an instruction type of the respective individual instruction associated
15 with the respective decision point and that identifies the respective
16 decision point; and

17 means for injecting a call at least one module that is capable
18 of evaluating the respective decision point with regard to whether a
19 failure is to be induced.

20
21 **23.** The arrangement as recited in claim 20, further comprising:
22 common language runtime means for executing the instrumented common
23 intermediate language code and the decision means in a runtime environment.

1 **24.** The arrangement as recited in claim 23, wherein the decision means
2 operates while the instrumented common intermediate language code is being
3 executed when the common language runtime means calls the decision means at
4 each decision point of the plurality of decision points.

5
6 **25.** The arrangement as recited in claim 20, wherein the decision means
7 comprises:

8 evaluation means for evaluating whether to throw an exception
9 responsive to a respective bookmark entry of each respective decision point
10 of the plurality of decision points and based on at least one throw exception
11 decision logic factor.

12
13 **26.** The arrangement as recited in claim 20, wherein the arrangement
14 comprises at least one device.

15
16 **27.** The arrangement as recited in claim 20, wherein the arrangement
17 comprises one or more processor-accessible media.

1 28. One or more processor-accessible media comprising an
2 instrumentation tool that is capable of determining whether respective instructions
3 from common intermediate language code meet at least one predetermined
4 exception-related criterion and that is adapted to inject respective decision points
5 into the common intermediate language code in association with the respective
6 instructions that meet the at least one predetermined exception-related criterion,
7 each injected respective decision point including an indication of an instruction
8 type of the respective associated instruction, an identifier of the injected respective
9 decision point, and a call to a program that can selectively cause an exception to
10 be thrown with respect to the respective associated instruction.

11
12 29. A method for instrumentation injection with regard to a common
13 language runtime environment, the method comprising:

14 determining whether an instruction from common intermediate language
15 code is capable of causing an exception; and

16 if so, injecting a decision point in association with the instruction to mark
17 the instruction for evaluation during a common language runtime execution, the
18 evaluation involving a decision as to whether a failure is to be induced with
19 respect to the instruction.
20
21
22
23
24
25

1 **30.** The method as recited in claim 29, wherein the determining
2 comprises:

3 determining whether the instruction from the common intermediate
4 language code is capable of causing an exception and is (i) related to a pre-
5 selected exception category and/or (ii) of a pre-selected instruction type.
6

7 **31.** The method as recited in claim 29, wherein the injecting comprises:
8 injecting an indicator of an instruction type of the instruction; and
9 injecting an identifier of the decision point.
10

11 **32.** The method as recited in claim 29, wherein the injecting comprises:
12 injecting a call to at least one module of a decision runtime library
13 that is adapted to perform the evaluation.
14

15 **33.** One or more processor-accessible media comprising processor-
16 executable instructions that, when executed, direct a device to perform the method
17 as recited in claim 29.
18

19 **34.** The method as recited in claim 29, further comprising:
20 repeating the determining and the injecting for a plurality of instructions
21 from the common intermediate language code; and
22 producing instrumented common intermediate language code as a result of
23 the repeating.
24
25

1 **35.** The method as recited in claim 34, further comprising:
2 detecting the decision point in the instrumented common intermediate
3 language code during execution thereof; and
4 calling at least one module of a decision runtime library, which is adapted
5 to perform the evaluation, as a result of the detecting.

6
7 **36.** The method as recited in claim 29, further comprising:
8 selectively deciding whether the execution is to fail at the decision point.

9
10 **37.** The method as recited in claim 36, further comprising:
11 if it is decided at the selectively deciding that the execution is to fail at the
12 decision point, then choosing which exception of at least two exceptions is to be
13 thrown.

14
15 **38.** The method as recited in claim 36, further comprising:
16 if it is decided at the selectively deciding that the execution is to fail at the
17 decision point, then inducing a failure in the execution of the common language
18 runtime with respect to the instruction.